

TEMAS

Mongo



Express



Angular



Node



PHP



JavaScript



HTML5



CSS3



Sublime Text



Bootstrap



Grunt



Vagrant



Docker



PATROCINADORES

Estructura de datos en C - Árboles



@telecristy

Publicado
02/Jul/2013

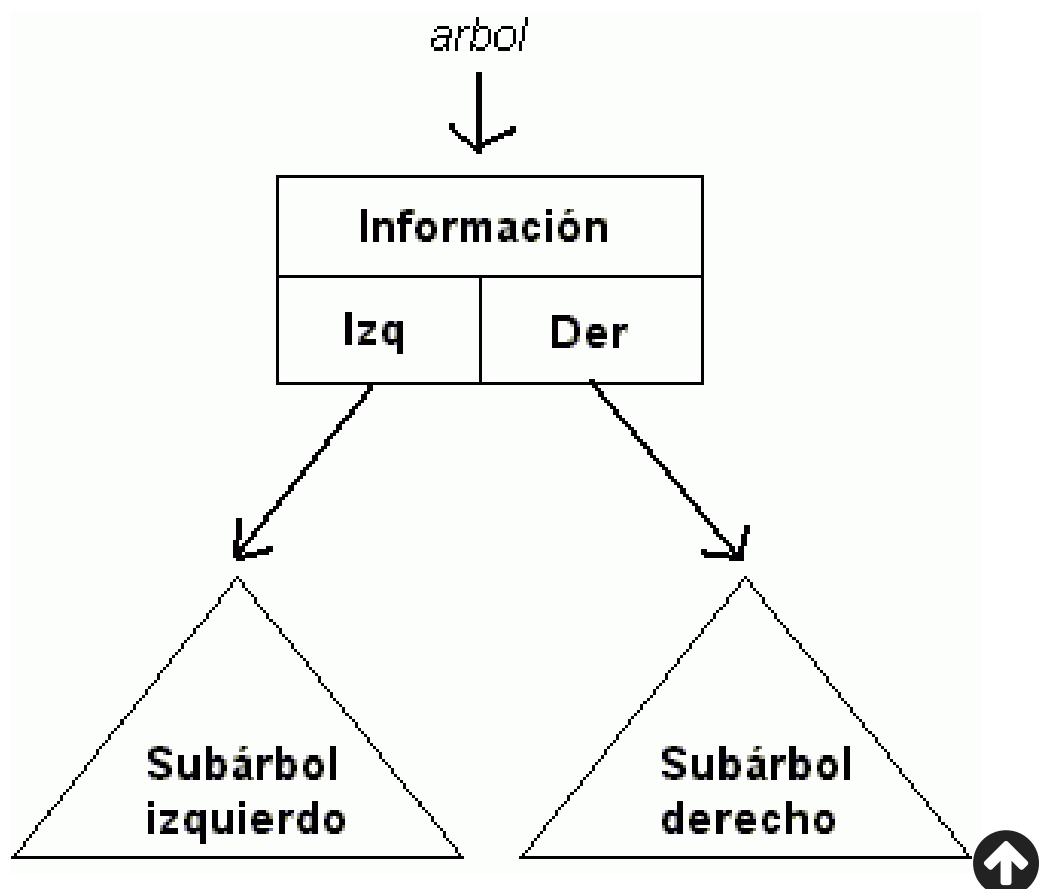
f COMPARTIR EN FACEBOOK

t COMPARTIR EN TWITTER

g+ COMPARTIR EN GOOGLE+

Esta publicación tiene 47 meses de antigüedad, te invitamos a buscar publicaciones más recientes relacionados a este tema [aquí](#).

Las listas ligadas, las pilas y las colas son estructuras de datos lineales. Un árbol es una estructura de datos no lineal de dos dimensiones, con sus propiedades especiales. Tres nodos contienen dos o más ligas.



Explicaremos los árboles binarios; árboles cuyos nodos contienen dos ligas (ninguna, una, o ambas de las cuáles pueden ser NULL). El nodo raíz es el primer nodo del árbol. Cada liga del nodo raíz hace referencia a un hijo. El hijo izquierdo es el primer nodo del subárbol izquierdo, y el hijo derecho es el primer nodo del subárbol derecho. (Ver imagen).

Investigando sobre el tema encontré que los científicos en computación generalmente dibujan árboles del nodo raíz hacia abajo; exactamente lo contrario a un árbol natural.

En esta publicación vamos a crear un árbol binario de búsqueda. Un árbol binario de búsqueda (sin valores duplicados de nodos) tiene la característica de que los valores de cualquier subárbol izquierdo son menores que el valor de su nodo padre, y que los valores de cualquier subárbol derecho son mayores que el valor de su nodo padre.

A continuación les dejo el código, reciban un saludo de mi parte @telecristy.

```
/**
 * Crea un arbol binario y lo recorre en
 * preorden, inorden, y en postOrden
 */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

/* estructura autoreferenciada */
struct nodoArbol {
    struct nodoArbol *ptrIzq;
    int dato;
    struct nodoArbol *ptrDer;
};

typedef struct nodoArbol NodoArbol; /* sinónimo de la estructura nodoArbol */
typedef NodoArbol *ptrNodoArbol; /* sinónimo de NodoArbol* */

/* prototipos */
void insertaNodo(ptrNodoArbol *ptrArbol, int valor);
void inOrden(ptrNodoArbol ptrArbol);
void preOrden(ptrNodoArbol ptrArbol);
void postOrden(ptrNodoArbol ptrArbol);

/* la función main comienza la ejecución del programa */
int main()
{
    int i; /* contador para el ciclo de 1 a 10 */
    int elemento; /* variable para almacenar valores al azar */
    ptrNodoArbol ptrRaiz = NULL; /* árbol inicialmente vacío */

    srand(time(NULL));
    printf("Los números colocados en el arbol son:n");

    /* inserta valores al azar entre 1 y 15 en el árbol */
    for (i = 1; i <= 10; i++) {
        elemento = rand() % 15;
        printf("%3d", elemento);
        insertaNodo(&ptrRaiz, elemento);
    }

    /* recorre el árbol en preorden */
    printf("\nnEl recorrido en preorden es:n");
    preOrden(ptrRaiz);

    /* recorre el árbol en inorden */
    printf("\nnEl recorrido inorden es:n");
    inOrden(ptrRaiz);

    /* recorre el árbol en postOrden */
    printf("\nnEl recorrido en postOrden es:n");
    postOrden(ptrRaiz);

    return 0;
}

/* inserta un nodo dentro del árbol */
void insertaNodo(ptrNodoArbol *ptrArbol, int valor)
{
    /* si el árbol está vacío */
    if (*ptrArbol == NULL) {
        *ptrArbol = malloc(sizeof(NodoArbol));

        /* si la memoria está asignada, entonces asigna el dato */
        if (*ptrArbol != NULL) {
            (*ptrArbol)->dato = valor;
            (*ptrArbol)->ptrIzq = NULL;
            (*ptrArbol)->ptrDer = NULL;
        } else {
            printf("no se inserto %d. No hay memoria disponible.n", valor);
        }
    } else {
        /* el dato a insertar es menor que el dato en el nodo actual */
        if (valor < (*ptrArbol)->dato) {
            insertaNodo(&(*ptrArbol)->ptrIzq, valor);
        } else if (valor > (*ptrArbol)->dato) {
            insertaNodo(&(*ptrArbol)->ptrDer, valor);
        } else {
            printf("dup");
        }
    }
}
```

```

}
}
}

/* comienza el recorrido inorden del árbol */
void inOrden(ptrNodoArbol ptrArbol)
{
    /* si el árbol no está vacío, entonces recórrelo */
    if (ptrArbol != NULL) {
        inOrden(ptrArbol->ptrIzq);
        printf("%3d", ptrArbol->dato);
        inOrden(ptrArbol->ptrDer);
    }
}

/* comienza el recorrido preorden del árbol */
void preOrden(ptrNodoArbol ptrArbol)
{
    /* si el árbol no está vacío, entonces recórrelo */
    if (ptrArbol != NULL) {
        printf("%3d", ptrArbol->dato);
        preOrden(ptrArbol->ptrIzq);
        preOrden(ptrArbol->ptrDer);
    }
}

/* comienza el recorrido postOrden del árbol */
void postOrden(ptrNodoArbol ptrArbol)
{
    /* si el árbol no está vacío, entonces recórrelo */
    if (ptrArbol != NULL) {
        postOrden(ptrArbol->ptrIzq);
        postOrden(ptrArbol->ptrDer);
        printf("%3d", ptrArbol->dato);
    }
}

```

Saludos.



¿Te gustó esta publicación? Márcala como favorita ★



@telecristy

@telecristy

Estados Unidos

<https://www.codejobs.biz/>

Publicaciones del autor

Favoritos del autor

ARTÍCULOS RELACIONADOS



53 Javascript - Diferentes tipos de...



Cómo desarrollar una agenda telef...



Notación Húngara



Aprende Laravel desde cero: Instala...



27 Javascript - Objeto número



¿Qué es ITIL?



4 Comments

Codejobs

Login

Recomendar

Compartir

ordenar por el mejor



Únete a la discusión...



Brayme Guaman · hace 4 meses

falta un cast quedaria así:

```
*ptrArbol = (ptrNodoArbol)malloc(sizeof(NodoArbol));
```

^ | v · Reply · Compartir



Luis Ricardo Martinez · hace un año

Hola, buen codigo pero me podrias explicar que significa el printf("dup \n"); ?

^ | v · Reply · Compartir



Jose Ignacio Jimenez Aranguiz → Luis Ricardo Martinez · hace 5 meses

Cuando el valor que ingresa es igual al que esta en el nodo se toma como duplicado, y puso ese 'dup' para saber que era el mismo numero :D

^ | v · Reply · Compartir



Quique Castro → Jose Ignacio Jimenez Aranguiz · hace 9 días

sabes como funciona el puntero con el mismo nombre de la estructura?

^ | v · Reply · Compartir

TAMBIÉN EN CODEJOBS

Siguiendo instrucciones de Stack Overflow - Aprende a Programar

1 comment · hace un año



Fredy — al margen de StackOverflow, eso pasa cuando copian y pegan sin compilar en el C:/erebro: entendiendo lo que están desarrollando. saludos

Curso: Aprende React.js desde cero #1 - Aprende a Programar

2 comments · hace 4 días



Cudox — lo ejecuto por git bash y me sale lo siguiente
https://uploads.disquscdn.c...

JavaScript es fácil: ¿Qué es el Event Delegation? #2 - Aprende a Programar

1 comment · hace un año



Eruz Apodaca — mindfuck del minuto 0:30 al 0:50

Instalar el tema Seti_UI para Sublime Text 3 - Aprende a Programar

3 comments · hace un año

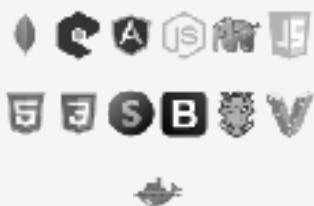


Lucas Corso — Muy bueno el tema, gracias.

[Subscribe](#) [Agrega Disqus a tu sitio](#) [Add Disqus](#) [Add](#) [Privacidad](#)

DISQUS

Tecnologías



Codejobs es una comunidad de amantes a la tecnología que creemos que el conocimiento es la única forma de ser verdaderamente libres y autónomos.



© 2017 Codejobs

